FEO3260 — Fundamentals of Machine Learning Networks

CA4: Yusen Wang, Li Cheng, Yifei Dong, Jeannie He

2023, March 21

1 Decentralized gradient descent with 10 workers

In the master-worker computing structure, the initial dataset is partitioned into 10 randomly disjoint segments of equal length. Each worker sensor node performs the parallel computation of their respective gradients, which are subsequently transmitted to the master node. The master node utilizes all sub-gradient information to update the weight w_{k+1}. The topology of the computing structure is shown in Fig. 1.

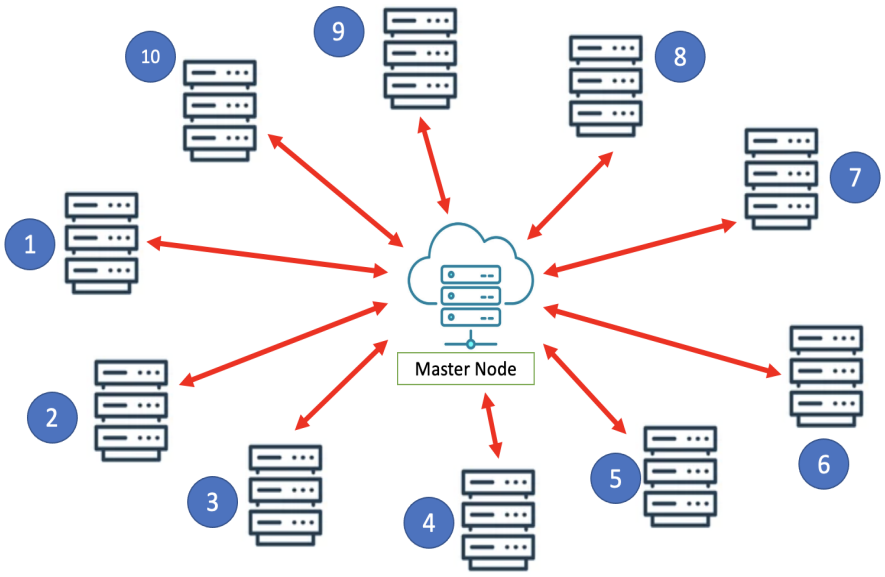

Figure 1: Master-worker structure.

1.1 Characterize the convergence against p and R

The convergence against probability p (with $R = 1$) is shown in Fig. 2

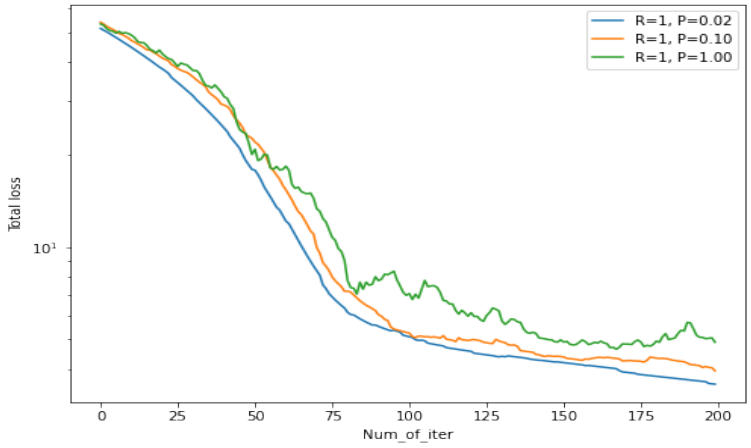

Figure 2: Convergence against p.

The convergence against variance R (with $P = 0.5$) is shown in Fig. 3

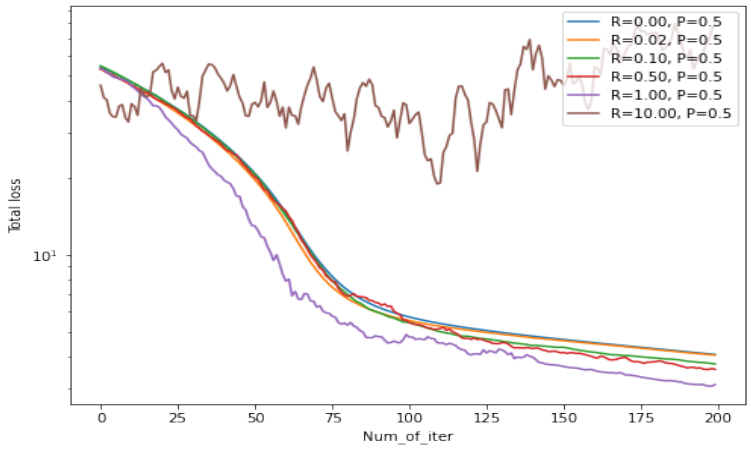

Figure 3: Convergence against R.

1.2 Propose an efficient method to improve the robustness

In the modified approach to improve the robustness of the decentralized gradient descent algorithm against added Gaussian noise, we propose to include a noise reduction step during the gradient computation. Specifically, the gradient computation step at each worker node is modified to include a noise reduction function that subtracts the average of the gradients from all worker nodes at that iteration. This helps to eliminate the effect of the added Gaussian noise on the gradient computation.

The noise reduction function can be defined as follows:

Let $\nabla f(w_k)$ denote the gradient of the objective function at worker i at iteration k, The noise reduction function subtracts the average of the gradients from all worker nodes at that iteration from each individual gradient:

$$\nabla f'(w_k) = \nabla f(w_k) - \frac{1}{N}\sum_{j=1}^{N} \nabla f_j(w_k) \tag{1}$$

where N is the total number of worker nodes.

Intuitively, the noise reduction function works by estimating the true gradient by removing the added Gaussian noise. By taking the average of the gradients from all worker nodes, we obtain an estimate of the true gradient, and subtracting this estimate from each individual gradient helps to eliminate the added noise.

2 Two-star Topology

Consider a two-star topology with communication graph (1,2,3,4)-5-6-(7,8,9,10), the topology is shown in Fig. 4.

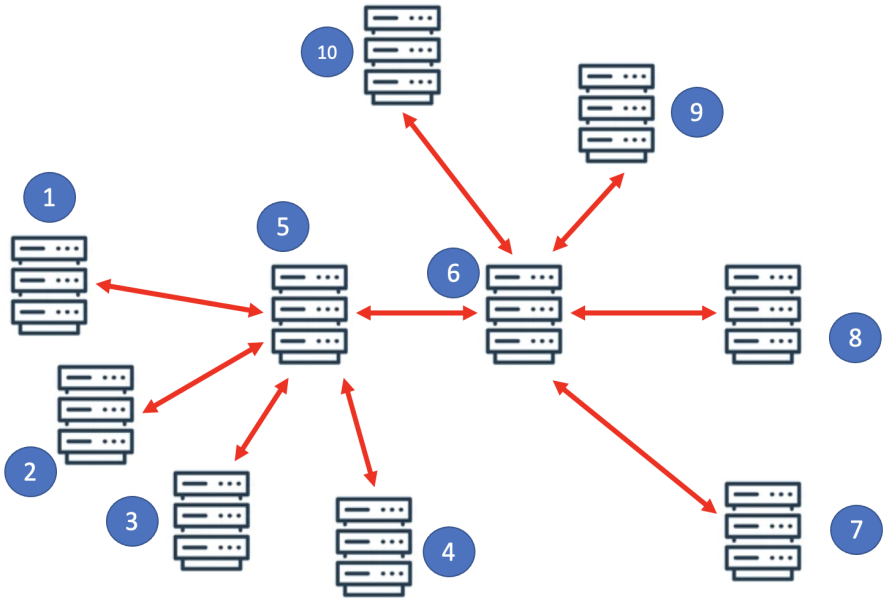

Figure 4: Two-star structure.

2.1 Characterize the convergence against p and R

The convergence against probability p (with $R = 1$) is shown in Fig. 5

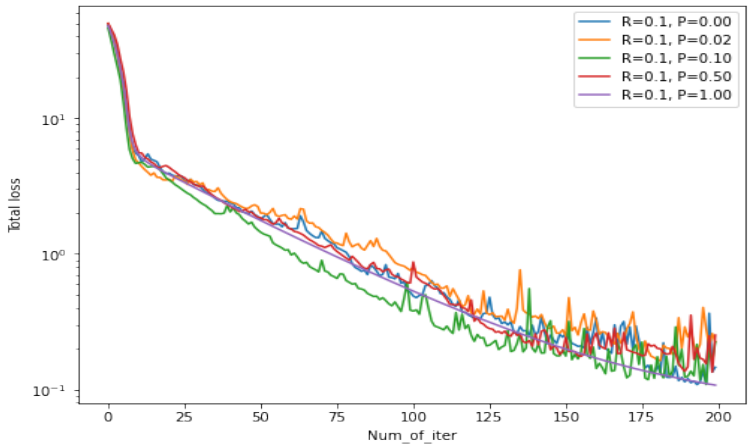

Figure 5: Convergence against p.

The convergence against variance R (with $P = 0.5$) is shown in Fig. 6

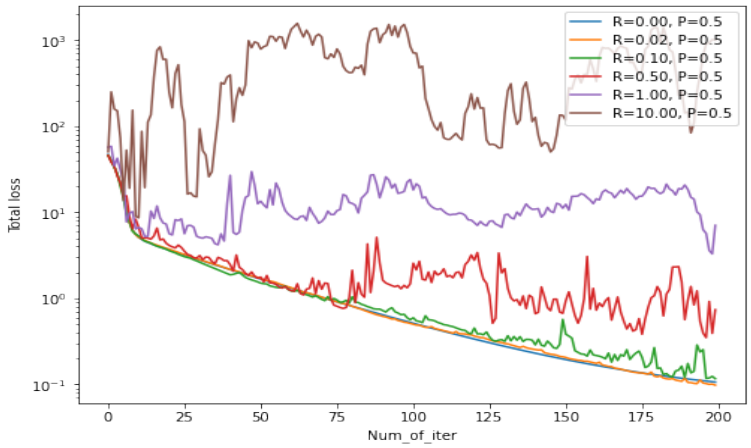

Figure 6: Convergence against R.

2.2 Propose an efficient method to improve the robustness

To improve the robustness of the decentralized gradient descent algorithm for the two-star topology with communication graph (1,2,3,4)-5-6-(7,8,9,10), we can modify the algorithm as follows:

For k = 1,2,..., do the following

1. Each worker node i computes its gradient $\nabla f_i(w_k)$ with bias and adds a zero-mean Gaussian noise with a large variance R with probability P.

2. Each worker node i sends its gradient $\nabla f_i(w_k)$ to its corresponding hub node 5 or 6.

3. Each hub node 5 or 6 computes the average of the received gradients from its worker nodes and broadcasts the average to all its neighbor nodes in the communication graph.

4. Each worker node i receives the averaged gradient from its corresponding hub node and computes the corrected gradient $\nabla f_i'(w_k)$ by subtracting the estimate of the true gradient (the average gradient across all worker nodes) from its individual gradient computation

5. Each worker node i updates its weight parameter using the corrected gradient $\nabla f_i'(w_k)$

6. The algorithm continues until convergence criteria are met.

The modification involves adding a bias correction step to each worker node's gradient computation, as well as averaging the gradients at the hub nodes before broadcasting them to the neighboring nodes. This helps to reduce the impact of added Gaussian noise and ensure that the weight updates are accurate.

3 Protect Three Workers

In Algorithm 1, since all workers are assumed to have the same probability of adding noise, we should aim to protect the three workers whose contributions to the overall gradient are the most significant. One approach to identify these workers is to compute the magnitudes of their individual gradients, select the three workers with the highest magnitudes, and ensure they are protected.

In the two-star topology, protecting the hub nodes is crucial as they receive and broadcast the gradients from multiple worker nodes, affecting the overall gradient computation. Thus, we should protect nodes 5, 6, and a node with the highest gradient magnitude.